น.พ.สมบัติ เตรียมแจ้งอรุณ

SOFT MAIL

หนูน้อยมหัศจรรย์

รายละเอียดการใช้งานโปรแกรมควบคุมเมาส์และฟังก์ชันที่ใช้บ่อย ๆ ร่วมกับพูลดาวน์เมนู

ในปัจจุบัน เราจะพบเห็นโปรแกรมสำเร็จรูปทางด้านกราฟิกมากมาย และสิ่งหนึ่งซึ่งมักจะขาดไม่ได้ก็คือเมาส์ คงทราบแล้วนะครับว่าหนูน้อยมหัศจรรย์ของผู้เขียนก็คือเมาส์นั่นเอง ผู้เขียนมักถูกถามเสมอว่าใช้จอโมโนโครมอยู่ ทำซื้อเมาส์มาจะคุ้มค่าหรือเปล่า อันที่จริงแล้วเราสามารถเขียนโปรแกรมใช้เมาส์ของเราเองได้โดยไม่ยากเลย ในฉบับนี้ผู้เขียนขอเสนอโปรแกรมการควบคุมเคอร์เซอร์ และสร้างพูลดาวน์เมนูในโหมดกราฟิก โดยการใช้แป้นพิมพ์ร่วมกับการใช้เมาส์บนจอโมโนโครม และใช้เทอร์โบปาสคาล เวอร์ชัน 5.0

เรามาดูโปรแกรมการควบคุมเมาส์กันก่อนดีกว่า (ดูรายละเอียดของโปรแกรมในรูปที่ 1 ประกอบ) ในที่นี้จะหมายถึงตามมาตรฐานของไมโครซอฟต์เมาส์ เมาส์นั้นมีฟังก์ชันให้ใช้มากมาย แต่มีฟังก์ชันสำคัญที่ใช้งานบ่อย ๆ ดังนี้

Function reset__m: boolean;

- ใช้ตรวจสอบว่ามีการใช้เมาส์หรือไม่

Function lpress: boolean;

- ใช้ตรวจสอบว่ามีการกดปุ่มซ้ายของเมาส์หรือไม่

Function rpress: boolean;

- ใช้ตรวจสอบว่ามีการกดปุ่มขวาของเมาส์หรือไม่

Function cursor__move: boolean;

- ใช้ตรวจสอบว่ามีการเคลื่อนเมาส์จากตำแหน่งเดิมหรือไม่

Procedure xrange (x,y: integer);

- ใช้กำหนดช่วงกว้างในแนวนอน

Procedure yrange (x,y: integer);

- ใช้กำหนดช่วงกว้างในแนวตั้ง

Procedure getcursor (var x,y: integer);

- ใช้อ่านค่าตำแหน่งปัจจุบันของเมาส์

รูปที่ 1 แสดงรายละเอียดของโปรแกรม MOUSE.PAS

```
program showmouse;
uses crt,dos,graph;
type
  menustr   = array[1..10] of string[10];
var
  graphdriver,graphmode,
  fnum,arg2,arg3,arg4,
  x1,y1,x,y,z,
  p,q,r,s,choice              : integer;
  menumap,blockmap,curmap     : pointer;
  menulist                    : menustr;
  ch,chtype                   : char;
  loop,loop1,loop2,mouses     : boolean;
  regs                        : registers;

procedure HercDriverProc; external;
{$L HERC.OBJ }

procedure TriplexFontProc; external;
{$L TRIP.OBJ }

procedure keycheck(ch : char;var chtype : char);
var
  ch1 : char;
begin
  if ord(ch) = 0 then
  begin
    ch1 := readkey;
    case ch1 of
    #72 : chtype := 'U'; {Up}     #75 : chtype := 'L'; {Left}
    #77 : chtype := 'R'; {Right}  #80 : chtype := 'D'; {Down}
    #71 : chtype := 'H'; {Home}   #79 : chtype := 'E'; {End}
    #73 : chtype := 'P'; {PgUp}   #81 : chtype := 'p'; {PgDn}
    #68 : chtype := '0'; {F10}
    else
     chtype := ' ';
    end;
  end
  else
    chtype := 'C';
end;

procedure movleftright(var x : integer;scale : integer);
begin
  x := x+scale;
  if x > 719 then x := 719;
  if x < 0 then x := 0;
end;

procedure movupdown(var y : integer;scale : integer);
begin
  y := y+scale;
  if y > 347 then y := 347;
  if y < 0 then y := 0;
end;

procedure cursor(x,y : integer);
begin
   line(x-5,y,x+5,y);line(x,y-3,x,y+3);
   putpixel(x-4,y,0);putpixel(x+4,y,0);putpixel(x,y-2,0);putpixel(x,y+2,0);
end;

procedure movcursor(var x,y : integer;scale1,scale2 : integer);
begin
```

รูปที่ 1 (ต่อ)

```
  movupdown(y,scale2);movleftright(x,scale1);
end;

procedure mouse(var a,b,c,d : integer);
begin
  regs.AX := a;regs.BX := b;regs.CX := c;regs.DX := d;
  intr($33,regs);
  a := regs.AX;b := regs.BX;c := regs.CX;d := regs.DX;
end;

function reset_m: boolean;
begin
  fnum := 0;
  mouse(fnum,arg2,arg3,arg4);
  if fnum <> 0 then reset_m := true else reset_m := false;
end;

procedure xrange(x,y : integer);
begin
  fnum := 7;
  mouse(fnum,arg2,x,y);
end;

procedure yrange(x,y : integer);
begin
  fnum := 8;
  mouse(fnum,arg2,x,y);
end;

procedure setcursor(x,y : integer);
begin
  fnum := 4;
  mouse(fnum,arg2,x,y);
end;

function lpress: boolean;
begin
  fnum := 3;
  mouse(fnum,arg2,arg3,arg4);
  if arg2 = 1 then lpress := true else lpress := false;
end;

function rpress: boolean;
begin
  fnum := 3;
  mouse(fnum,arg2,arg3,arg4);
  if arg2 = 2 then rpress := true else rpress := false;
end;

function cursor_move: boolean;
begin
  fnum := 11;
  mouse(fnum,arg2,arg3,arg4);
  if ((arg3 = 0) and (arg4 = 0)) then
     cursor_move := false
  else
     cursor_move := true;
end;

procedure getcursor(var x,y : integer);
begin
  fnum := 3;
  mouse(fnum,arg2,x,y);
end;
```

รูปที่ 1 (ต่อ)

```
procedure show(x : integer);
begin
  case x of
  1..9 : outtextxy(x1,y1,menulist[x]);
  10   : begin
           closegraph;halt(1);
         end;
  end;
end;

function menu(var choice : integer): boolean;
var
  choicemap                : pointer;
  choicesize,x,y,z,xx,yy : integer;
  loop2                    : boolean;
begin
  menu := true;
  if mouses then
  begin
     getcursor(xx,yy);
     setcursor(0,(choice-1)*20);
  end;
  choicesize := imagesize(5,0,118,20);
  getmem(choicemap,choicesize);
  getimage(0,0,120,210,blockmap^);
  putimage(0,0,menumap^,copyput);
  z := choice;
  getimage(5,5+((choice-1)*20),115,5+(choice*20),choicemap^);
  putimage(5,5+((choice-1)*20),choicemap^,notput);
  loop2 := true;
  while loop2 do
  begin
    if choice <> z then
    begin
      putimage(5,5+((choice-1)*20),choicemap^,copyput);
      getimage(5,5+((z-1)*20),115,5+(z*20),choicemap^);
      putimage(5,5+((z-1)*20),choicemap^,notput);
      choice := z;
    end;
    if mouses then
    begin
      if cursor_move then
      begin
        getcursor(x,y);
        z := (y div 20)+1;
      if z > 10 then
      begin
        z := 10;y := 200;setcursor(x,y);
      end;
    end;
    if lpress then
    begin
      repeat until lpress = false;
      loop2 := false;
    end;
    if rpress then
    begin
      repeat until rpress = false;
      menu := false;loop2 := false;
    end;
  end;
  if keypressed then
  begin
    ch := readkey;
    keycheck(ch,chtype);
    case chtype of
```

รูปที่ 1 (ต่อ)

```
        'U' : if z = 1 then z := 10 else z := z-1;
        'D' : if z = 10 then z := 1 else z := z+1;
        'C' : begin
                ch := upcase(ch);
                case ch of
                #13 : loop2 := false;
                #27 : begin
                        menu := false;loop2 := false;
                      end;
                else
                  begin
                    x := 1;
                    while (ch <> menulist[x][1]) and (x < 11) do inc(x);
                    if x < 11 then
                    begin
                      choice := x;loop2 := false;
                    end;
                  end;
                end;
              end;
      end;
    end;
  end;
  putimage(0,0,blockmap^,copyput);
  freemem(choicemap,choicesize);
  if mouses then setcursor(xx,yy);
end;

begin
  fillchar(menulist,sizeof(menulist),#0);
  menulist[1] := 'BLOCK';        menulist[2] := 'CIRCLE';
  menulist[3] := 'DRAW';         menulist[4] := 'ELLIPSE';
  menulist[5] := 'FLOOD';        menulist[6] := 'LINE';
  menulist[7] := 'POSITION';     menulist[8] := 'RECTANGLE';
  menulist[9] := 'SCREEN';       menulist[10] := 'QUIT';
  if reset_m then
  begin
    xrange(0,719);yrange(0,347);
    setcursor(360,174);mouses := true;
  end
  else
    mouses := false;
  x1 := 360;y1 := 174;
  if RegisterBGIdriver(@HercDriverProc)  < 0 then halt(1);
  if RegisterBGIfont(@triplexfontproc) < 0 then halt(1);
  graphdriver := detect;
  initgraph(graphdriver,graphmode,'');
  p := imagesize(0,0,10,6);
  getmem(curmap,p);
  setlinestyle(0,0,3);
  settextjustify(lefttext,toptext);
  p := imagesize(0,0,120,210);
  getmem(menumap,p);
  getmem(blockmap,p);
  setactivepage(1);
  rectangle(1,1,119,209);
  settextstyle(1,0,1);
  for x := 1 to 10 do
      outtextxy(5,5+(x-1)*20,menulist[x]);
  getimage(0,0,120,210,menumap^);
  setactivepage(0);
  choice := 1;
  setlinestyle(0,0,1);
  settextjustify(lefttext,bottomtext);
```

รูปที่ 1 (ต่อ)

```
        loop := true;
        while loop do
        begin
          if x1-5 < 0 then p := 0 else p := x1-5;
          if y1-3 < 0 then q := 0 else q := y1-3;
          if x1+5 > 719 then r := 719 else r := x1+5;
          if y1+3 > 347 then s := 347 else s := y1+3;
          getimage(p,q,r,s,curmap^);
          cursor(x1,y1);
          loop1 := true;
          while loop1 do
          begin
            if keypressed then
            begin
              loop1 := false;
              ch := readkey;
              putimage(p,q,curmap^,copyput);
              keycheck(ch,chtype);
              case chtype of
              '0': begin
                     if menu(choice) then show(choice);
                     if mouses then setcursor(x1,y1);
                   end;
              'C': begin
                     if ch in ['1'..'9'] then
                     begin
                       case ch of
          '1' : movcursor(x1,y1,-1,1);     '3' : movcursor(x1,y1,1,1);
          '7' : movcursor(x1,y1,-1,-1);    '9' : movcursor(x1,y1,1,-1);
          '2' : movupdown(y1,1);           '4' : movleftright(x1,-1);
          '6' : movleftright(x1,1);        '8' : movupdown(y1,-1);
                     end;
                   end;
                 end;
              'H': movcursor(x1,y1,-15,-9);
              'E': movcursor(x1,y1,-15,9);
              'U': movupdown(y1,-9);
              'D': movupdown(y1,9);
              'L': movleftright(x1,-15);
              'R': movleftright(x1,15);
              'P': movcursor(x1,y1,15,-9);
              'p': movcursor(x1,y1,15,9);
              else
                loop1 := true;
              end;
              if mouses then setcursor(x1,y1);
            end;
            if mouses then
            begin
              if cursor_move then
              begin
                putimage(p,q,curmap^,copyput);
                getcursor(x1,y1);loop1 := false;
              end;
              if lpress then
              begin
                putimage(p,q,curmap^,copyput);
                repeat until lpress = false;
                if menu(choice) then show(choice);
                loop1 := false;
              end;
            end;
          end;
        end;
      end;
    end.
```

Procedure setcursor
(x,y: interger);

- ใช้กำหนดค่าตำแหน่งของเมาส์

Procedure mouse
(var a,b,c,d: integer);

- ใช้เรียกฟังก์ชันต่าง ๆ ของเมาส์ ซึ่งในโปรแกรมนี้จะเรียกโดยกำหนดหมายเลขฟังก์ชันให้ตัวแปร fnum

การทำงานของโปรแกรมจะเริ่มโดยการกำหนดตัวเลือกของเมนูไว้ในตัวแปร menulist จากนั้นจะทำการตรวจสอบว่ามีการใช้เมาส์หรือไม่ โดยใช้ฟังก์ชัน reset_m ถ้ามีการใช้เมาส์ก็จะกำหนดขนาดความกว้างในแนวนอนและแนวตั้ง กำหนดตำแหน่งกลางจอภาพและให้ค่า True ในตัวแปร mouses ในโปรแกรมนี้ใช้ Triplex Font โดยจะเก็บทั้งเฮอร์คิวลิส ไดรเวอร์และฟอนต์ไว้ในตัวโปรแกรม โดยลิงค์กับ Herc.obj และ Trip.obj รายละเอียดผู้อ่านสามารถศึกษาจากโปรแกรมตัวอย่างของเทอร์โบปาสคาล จากนั้นโปรแกรมจะสร้างเมนูและเก็บเมนูในรูปของอิมเมจโดยใช้ตัวแปรพอยน์เตอร์ menumap เป็นตัวชี้ ตัวแปรทั่วไปที่สำคัญในโปรแกรมสรุปได้ดังนี้

- blockmap เป็นตัวแปรพอยน์เตอร์ที่ชี้อิมเมจของจอภาพที่ถูกเมนูเขียนทับ

- curmap เป็นตัวแปรพอยน์เตอร์ที่ชี้อิมเมจของจอภาพที่ถูกเคอร์เซอร์เขียนทับ

- x1,y1 เป็นค่าตำแหน่งตามแนวนอน และแนวตั้งของเคอร์เซอร์ตามลำดับ

- p,q,r,s เป็นค่าตำแหน่งตามแนวนอน และแนวตั้งของอิมเมจที่ curmap ชี้

- choice เป็นค่าตำแหน่งตัวเลือกในเมนู

- fnum, arg2, arg3, arg4, ใช้เรียกฟังก์ชันของเมาส์

หลังจาก Install เมาส์ และเรียกโปรแกรมทำงานแล้ว จะปรากฏเคอร์เซอร์อยู่กลางจอภาพ สามารถเลื่อนเคอร์เซอร์โดยใช้แป้นพิมพ์ Arrow, Home, End, Pgup, Pdgn เพื่อเลื่อนทีละหลายพิกเซล หรือใช้ตัวเลขเพื่อเลื่อนทีละพิกเซล หรือเลื่อนเมาส์เมื่อกดแป้น F10 หรือกดปุ่มซ้ายของเมาส์ จะปรากฏพูลดาวน์เมนูบริเวณกรอบบนซ้ายของจอภาพ (รูปที่ 2) สามารถเลือกตัวเลือกได้ 3 วิธีคือ

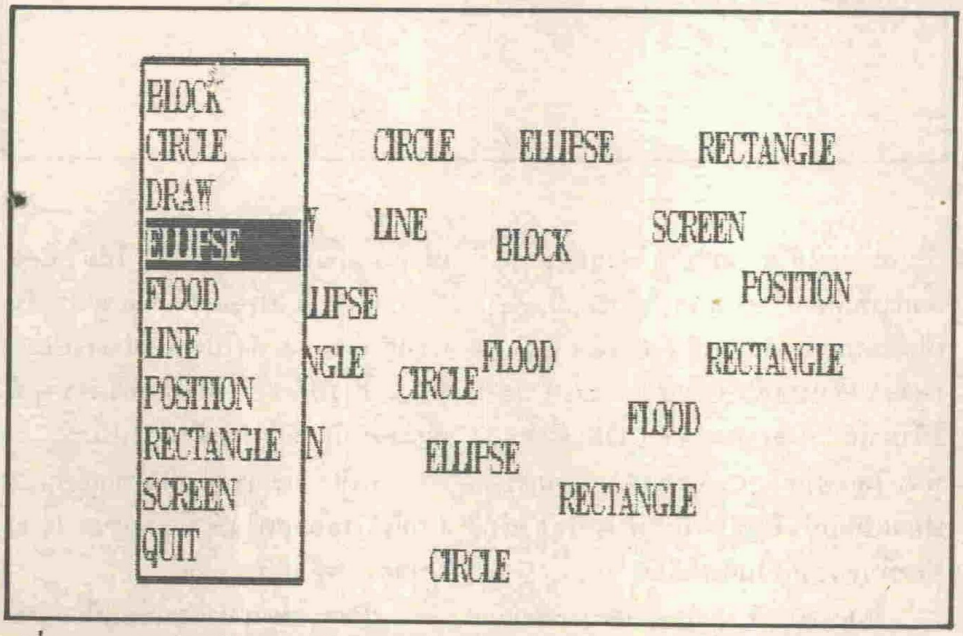
1. ใช้แป้นลูกศรขึ้น หรือลงไปยังตำแหน่งที่ต้องการ แล้วกดแป้น <ENTER>

2. กดแป้นตัวอักษรนำหน้าของตัวเลือกที่ต้องการ

3. เลื่อนเมาส์ขึ้น หรือลงไปยังตำแหน่งที่ต้องการ แล้วกดปุ่มซ้ายของเมาส์

ในกรณีที่จะไม่เลือกตัวเลือก กดแป้น <ESC> หรือกดปุ่มขวาของเมาส์ จอภาพจะแสดงเคอร์เซอร์ตามเดิม เมื่อเลือกตัวเลือกแล้วโปรแกรมจะเขียนตัวเลือกใน menulist ลงตรงตำแหน่งของเคอร์เซอร์ หรือออกจากโปรแกรมถ้าเลือก 'QUIT' ในส่วนนี้ก็เป็นหน้าที่ของผู้อ่านที่จะนำไปประยุกต์ใช้ในการเขียนโปรแกรมใช้งานตามต้องการ

ท้ายสุดนี้ ผู้เขียนหวังว่า โปรแกรมดังกล่าวคงจะช่วยให้เห็นคุณค่าของเมาส์ขึ้นบ้างไม่มากก็น้อย *สวัสดีครับ...*



รูปที่ 2 แสดงจอภาพขณะโปรแกรมถูกเรียกพูลดาวน์เมนู